Information Freshness and Popularity in Mobile Caching

Clement Kam*, Sastry Kompella*, Gam D. Nguyen*, Jeffrey E. Wieselthier† and Anthony Ephremides‡

*Information Technology Division, Naval Research Laboratory, Washington, DC
†Wieselthier Research, Silver Spring, MD
‡Electrical and Computer Engineering Department, University of Maryland, College Park, MD

***Abstract*—We propose a model for mobile caching in which the rate of requests for content is dependent on the *popularity* and the *freshness* of the information. We model popularity based on the history of requests and freshness based on the age of the content. We consider a discrete time (slotted) system in which new packets arrive at a limited capacity cache at discrete times. We prove that the optimal policy for choosing the set of packets to reside in a full cache when a packet arrives is to reject the one with the lowest request rate in that particular slot. Thus, there is no advantage to separately knowing the history of requests or the age of the content. Since the optimal policy depends on the profile of the request process, we also study the expected behavior of the request model. We provide a sufficient condition under which the change in the request rate goes to zero and provide some numerical examples that illustrate this behavior. We also consider a slight alteration to the model, in which only the recent history of requests is used for determining the request rate. In this case, we provide a sufficient condition for when the rate is equal to zero, which approximates the duration of requests for content.**

I. Introduction

Users are generating content now at a much faster rate than ever before, resulting in a significant strain on the global internet. Studies have shown that the volume of data generated from smartphones is set to exceed PC traffic as a whole by the year 2020 [1]. With the proliferation of mobile devices, traditional methods such as increasing the amount of bandwidth, or deploying more base stations are not expected to be able to accommodate the predicted traffic increase. In these circumstances, content caching [2] has been recognized as one of the most effective means of reducing delays and improving latency performance of online content and other internet applications. It has been widely recognized that, by bringing the content closer to users, caches have the potential to greatly reduce network bandwidth usage, server load, and perceived service delays.

Content caching comes in many forms. There is hierarchical caching [3], where caches are placed at different network levels, with institutional caches at the top level and client caches forming the bottom most level of the hierarchy. When a request is not satisfied by the client cache, it is redirected to the institutional caches. In a distributed cache scheme, there are no intermediate caches and it falls on the institutional caches to serve each others cache misses. Other higher caching structures have also been envisioned [4]. Furthermore, caching is an important tool in varied applications ranging from femtocells [5] to SDN [6]. While the objective of traditional caching is to reduce the retrieval delay experienced by users when they request a certain object over the network, it is not always the main focus. For example, Borst et al [7] showed that it is also important to focus on bandwidth minimization by maximizing the traffic volume served from cache.

In keeping with the growing trend for ubiquitous computing and the Internet of Everything (IoE), in which people, data, processes, and things connect to each other and the internet, future networks are going to be comprised of a large number of small nodes, with limited caching abilities. This is especially true in the dynamic world of tactical edge networks supported by the DoD, which will be dominated by connected tracking and telemetry, surveillance, and sensor type applications. Such an increasing diversity in service expectations advocates the need for content delivery infrastructures that focuses on information freshness among different applications and content classes. Existing policies for cache management rely on simple heuristics such as Least Recently Used (LRU) and Least Frequently Used (LFU) to replace the cached content with a new one. The LRU policy was analyzed in [8].

In this paper, we focus on the relationship between the popularity of the content, as reflected in the request rate, and the freshness of such content. We propose a dynamic cache management policy that tracks the age of the content and the history of requests to choose what content to cache, such that the number of requests for content not in cache is minimized. This is especially relevant for the IoE, where devices have limited buffers and will have to change cached content frequently, and more importantly, in which past objects are usually not requested.

Our contributions in this paper can be summarized as follows:

1) We propose a dynamic model for requesting content that depends on the freshness and popularity of the content. We propose a policy for managing content when the cache is at capacity and new content arrives, and we prove the optimality of the proposed policy.
2) We analyze the evolution of the request rate for two versions of the proposed model, and we provide sufficient conditions for the settling time of the request rate:

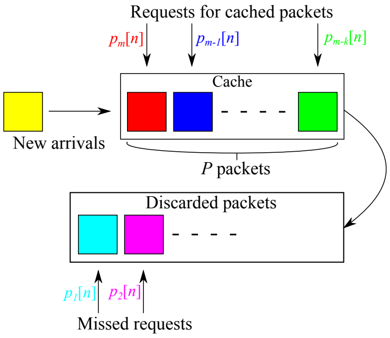

Fig. 1. Caching system model.

a) For the basic proposed model, we provide a sufficient condition for the time at which the change in the rate is equal to zero. This provides an approximation for when the requests reach a steady state.
b) In a slightly modified version of the model, only the recent history of requests (rather than the complete history) impacts the request rate, which guarantees in most cases that the request rate decays to zero. We present a sufficient condition for the time at which the request rate is equal to zero, which approximates the time duration for when a packet is requested.

II. System Model

We consider a discrete time (slotted) caching system (shown in Figure 1), in which there is a cache of size P that is sent new packets just prior to time slots $S_1, S_2, \ldots$. For a given policy π, a decision is made when a new packet is sent to a full cache, such that P of the $P+1$ packets among those in the cache and the new packet are selected to remain in the cache. For all packets that have ever been sent to the cache, whether they have been stored there or not, requests for the delivery of such content are made. The objective is to minimize the number of missed requests for packets (i.e., requests that occur when not in the cache) over some number of slots N.

The rate of requests is time-varying to reflect the popularity of files over time. The request process for packet m is given by $R_m[n]$, where $R_m[n] = 1$ if a request is made in slot n, and $R_m[n] = 0$ if no request is made in slot n. The probability of a request (or the request rate) for packet m in slot S_m is given by $p_m[S_m] \triangleq \Pr(R_m[S_m] = 1)$. The request rate evolves as

$$p_m[S_m + n] = (p_m[S_m] - \alpha n + \beta r_m([S_m, S_m + n - 1]))_0^1$$

where $r_m([S_m, S_m + n - 1])$ is the number of requests made for packet m in the interval $[S_m, S_m + n - 1]$, and $\alpha > 0$ and $\beta > 0$ are constant terms that weigh the effect of the packet's age n and history of requests $r_m([S_m, S_m + n - 1])$ on the request rate, respectively. The function $(x)_0^1$ is defined according to

$$(x)_0^a = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } 0 < x < a \\ a & \text{if } x \geq a \end{cases}$$

to ensure the probability is in $[0, 1]$. This request model reflects the following two assumptions. The first assumption is that packets with a higher age are less likely to be requested. The second is that packets that have been more popular (based on history of requests) continue to be more likely to be requested. We define $\tilde{p}_m[S_m + n] \triangleq p_m[S_m] - \alpha n + \beta r_m([S_m, S_m + n - 1])$ as the request value (i.e., $p_m[S_m + n] = (\tilde{p}_m[S_m + n])_0^1$).

Although we have the request model as a function of the two parameters of freshness and popularity, we are interested in a single metric that blends these two and is sufficient for effective cache management. We define the *effective age* of packet m in the cache as $\tilde{\Delta}_m[S_m + n] = \tilde{\Delta}_m[S_m] + n - \beta/\alpha \cdot r_m([S_m, S_m + n - 1])$, where $\tilde{\Delta}_m[S_m]$ is some initial effective age at the time of packet m's arrival to the cache. This statistic directly relates to the instantaneous request rate, and it conveys the idea that an effectively "fresher" packet is more desirable, and thus more frequently requested. The request rate can be expressed as $p_m[S_m + n] = (\alpha \tilde{\Delta}_m[S_m + n])_0^1$, where $\alpha \tilde{\Delta}_m[S_m] = p_m[S_m]$.

In the next section, we determine the optimal cache management policy for this specific request model. We have considered more generic request models, but as of yet have been unable to prove optimal policies in such cases, even in the simple case where the cache has a size of $P = 1$ packet.

III. Optimal Policy

We propose the following policy $\tilde{\pi}$ for selecting packets to remain in the cache when a packet arrives at a full cache: the P packets with the highest instantaneous request value remain in the cache. Formally, if a packet arrives at slot S and packet m has the instantaneous request value $\tilde{p}_m[S]$, we choose the set $\mathcal{C}[S] = \{m : \tilde{p}_m[S] \geq \min_m \tilde{p}_m[S], |\mathcal{C}[S]| = P\}$. We will show that this policy is optimal in the sense of minimizing the expected number of missed requests. This means that knowing the exact age and popularity of the packets is not necessary for selecting the optimal set of packets to cache. In the case where packets have the same initial rate of requests at their respective times of arrival, choosing the P packets with the lowest effective age is an equivalent policy to $\tilde{\pi}$.[1]

We denote the expected number of requests for packet m in the nth slot after slot S as $E[R_m[S+n]] = \Pr(R_m[S+n] = 1)$, and the total expected requests over the interval $[S, S+N]$ as $E[R_m^{[S,S+N]}] = \sum_{n=S}^{S+N} E[R_m[n]]$. We denote the event that there have been p requests for m in the interval $\mathcal{I}$ as $\varepsilon_{m,p}^{\mathcal{I}}$. Let $\mathcal{I}_n$ be the interval $[S, S+n]$.

Theorem 1: Policy $\tilde{\pi}$ minimizes the expected number of missed requests over the time duration N under consideration.

Proof: Let $\phi_m(n|q) \triangleq \Pr(R_m[S+n] = 1 | \varepsilon_{m,q}^{\mathcal{I}_{n-1}})$ be the probability of a request for packet m in slot $S+n$ given that $r_m([S, S+n]) = q$, $q \in \mathbb{Z}_{\geq 0}$. Since

$$\phi_m(n|q) = (\tilde{p}_m[S] - \alpha n + \beta q)_0^1,$$

[1] For this particular request model, the arrival process can be arbitrary as it does not affect the structure of the optimal policy.

it is easy to see that $\phi_{m_1}(n|q) \geq \phi_{m_2}(n|q)$ if $\tilde{p}_{m_1}[S] \geq \tilde{p}_{m_2}[S]$. We define the expected number of requests in a particular slot $S+n$ as

$$E[R_m[S+n]] = \sum_{q=0}^{n} \phi_m(n|q) \Pr(\varepsilon_{m,q}^{\mathcal{I}_{n-1}}) \tag{1}$$

We show that $E[R_{m_1}[S+n]] \geq E[R_{m_2}[S+n]]$ if $\tilde{p}_{m_1}[S] \geq \tilde{p}_{m_2}[S]$ as follows:

$$\begin{aligned}
&E[R_{m_1}[S+n]] - E[R_{m_2}[S+n]] \\
&\quad= \sum_{q=0}^{n} [\phi_{m_1}(n|q) \Pr(\varepsilon_{m_1,q}^{\mathcal{I}_{n-1}}) - \phi_{m_2}(n|q) \Pr(\varepsilon_{m_2,q}^{\mathcal{I}_{n-1}})] \\
&\quad\geq \sum_{q=0}^{n} [\phi_{m_1}(n|q)(\Pr(\varepsilon_{m_1,q}^{\mathcal{I}_{n-1}}) - \Pr(\varepsilon_{m_2,q}^{\mathcal{I}_{n-1}}))] \qquad (2) \\
&\quad\geq \phi_{m_1}(n|0) \sum_{q=0}^{n} [(\Pr(\varepsilon_{m_1,q}^{\mathcal{I}_{n-1}}) - \Pr(\varepsilon_{m_2,q}^{\mathcal{I}_{n-1}}))] \\
&\quad= 0 \qquad (3)
\end{aligned}$$

where (2) is from the the property of $\phi_m(n|q)$ for different $\tilde{p}_m[S]$ above, and (3) is from $\sum_{q=0}^{n} \Pr(\varepsilon_{m,q}^{\mathcal{I}_{n-1}}) = 1$. Since this holds for any $n \geq 0$, it also holds for the sum of slots from S to $S+N$. Therefore, choosing the P packets with the highest instantaneous rates at the start of the interval to reside in the cache and leaving out the one with the lowest instantaneous rate will yield the minimum number of missed requests. ■

In the case where every packet starts with the same $p_m[S_m]$ at its time of arrival to the cache S_m, the only term that matters for choosing the optimal set of packets to keep in the cache is $(-\alpha n + \beta r_m([S_m, S_m + n]))$. Equivalently, we can let each packet start with the same initial effective age at time of arrival $\tilde{\Delta}_m[S_m]$, and the effective age $\tilde{\Delta}_m[S_m + n]$ is all that is necessary to choose the optimal set of packets to remain in the cache.

IV. Request Model Analysis

Whether the instantaneous request value (or effective age) is a sufficient metric for optimal cache management depends on the specifics of the request model. We are interested in analyzing the behavior of the request model to get some insight into the performance of a particular policy. In this section, we study the time instant when, on average, the request rate stabilizes.

The request rate in slot $S+n$, on average, is the expression given in (1). The expression for the probability of there being q requests for packet m in the interval $\mathcal{I}_{n-1}$ can be defined iteratively as follows:

$$\begin{aligned}
\Pr(\varepsilon_{m,0}^{\mathcal{I}_0}) &= 1 - \phi_m(0|0) = 1 - (\tilde{p}_m[S])_0^1 \\
\Pr(\varepsilon_{m,1}^{\mathcal{I}_0}) &= \phi_m(0|0) = (\tilde{p}_m[S])_0^1.
\end{aligned}$$

For $n \geq 2$,

$$\Pr(\varepsilon_{m,q}^{\mathcal{I}_{n-1}}) = \begin{cases} \prod_{u=0}^{n-1}(1 - \phi_m(u|0)), & \text{if } q = 0 \\ \Pr(\varepsilon_{m,q}^{\mathcal{I}_{n-2}})(1 - \phi_m(n-1|q)) & \text{if } 1 \leq q \\ \quad + \Pr(\varepsilon_{m,q-1}^{\mathcal{I}_{n-2}})\phi_m(n-1|q-1), & \quad \leq n-1 \\ \prod_{u=0}^{n-1} \phi_m(u|u), & \text{if } q = n. \end{cases}$$

This can be shortened to

$$\begin{aligned}
\Pr(\varepsilon_{m,q}^{\mathcal{I}_{n-1}}) &= \Pr(\varepsilon_{m,q}^{\mathcal{I}_{n-2}})(1 - \phi_m(n-1|q)) \\
&\quad + \Pr(\varepsilon_{m,q-1}^{\mathcal{I}_{n-2}})\phi_m(n-1|q-1)
\end{aligned}$$

for $0 \leq q \leq n$ if we let $\Pr(\varepsilon_{m,q}^{\mathcal{I}_n}) \triangleq 0$ for $q < 0$ or $q > n+1$.

If $\alpha > \beta$, the request value $\tilde{p}_m[S+n]$ is strictly decreasing in n because $r_m([S, S+n-1])$ is non-decreasing in n. If $\alpha = \beta$, the request value is non-increasing in the extreme case where $r_m([S, S+n-1]) = n$ (i.e., there is a request in every slot), and will decrease in n on average if $\tilde{p}_m[S] < 1$. For the remainder of the paper, we focus on the case where $\alpha < \beta$, and the request rate can increase or decrease.

A. Change in Request Rate

We focus now on the change in the request rate from one slot to another, and we provide a sufficient condition for which the change in request rate goes to zero. We omit the subscript m in this section. The change in the request rate is given by

$$\begin{aligned}
&E[R[S+n+1]] - E[R[S+n]] \\
&= \sum_{q=0}^{n+1} \phi(n+1|q) \Pr(\varepsilon_q^{\mathcal{I}_n}) - \sum_{q=0}^{n} \phi(n|q) \Pr(\varepsilon_q^{\mathcal{I}_{n-1}}) \\
&= \sum_{q=0}^{n} [\phi(n+1|q)[\Pr(\varepsilon_q^{\mathcal{I}_{n-1}})(1 - \phi(n|q)) \\
&\quad + \Pr(\varepsilon_{q-1}^{\mathcal{I}_{n-1}})\phi(n|q-1)] - \phi(n|q) \Pr(\varepsilon_q^{\mathcal{I}_{n-1}})] \\
&\quad + \phi(n+1|n+1) \Pr(\varepsilon_n^{\mathcal{I}_{n-1}})\phi(n|n) \\
&= \sum_{q=0}^{n} (\phi(n+1|q) - \phi(n|q)) \Pr(\varepsilon_n^{\mathcal{I}_{n-1}}) \\
&\quad + \sum_{q=0}^{n} \phi(n+1|q)(\Pr(\varepsilon_{q-1}^{\mathcal{I}_{n-1}})\phi(n|q-1) \\
&\quad - \Pr(\varepsilon_q^{\mathcal{I}_{n-1}})\phi(n|q)) + \phi(n+1|n+1) \Pr(\varepsilon_n^{\mathcal{I}_{n-1}}) \\
&= \sum_{q=0}^{n} (\phi(n+1|q) - \phi(n|q)) \Pr(\varepsilon_n^{\mathcal{I}_{n-1}}) \\
&\quad + \sum_{q=0}^{n} (\phi(n+1|q+1) - \phi(n+1|q)) \Pr(\varepsilon_q^{\mathcal{I}_{n-1}}) \\
&\quad \times \phi(n|q) \\
&= \sum_{q=0}^{n} \Pr(\varepsilon_q^{\mathcal{I}_{n-1}})[(1 - \phi(n|q))\phi(n+1|q) \\
&\quad - \phi(n|q)(1 - \phi(n+1|q+1))]. \qquad (4)
\end{aligned}$$

For a given n, we define $q_0(n)$ to be the minimum value of $q \in [0, n]$ such that $\phi(n|q) > 0$. This can be found to be $\left(\left\lceil \frac{\alpha}{\beta} n - \frac{\tilde{p}[S]}{\beta} \right\rceil\right)_0^n$. For $n \leq \tilde{p}[S]/\alpha$, this yields $q_0(n) = 0$, but for larger n, $q_0(n) \geq 0$. For $q < q_0$, the change in request rate (4) evaluates to zero because $\phi(n|q) = 0$.

Similarly, for a given n, we define $q_1(n)$ to be the maximum value of $q \in [0, n]$ such that $\phi(n|q) < 1$. This can be found to be $\left(\left\lfloor \frac{\alpha}{\beta} n - \frac{\tilde{p}[S]}{\beta} + \frac{1}{\beta} \right\rfloor\right)_0^n$. For $n \leq (\beta + \tilde{p}[S] - 1)/\alpha$, this yields $q_1(n) = 0$, but for larger n, $q_1(n) \geq 1$. For $q > q_1(n)$, the change in request rate (4) evaluates to zero because $\phi(n|q) = 1$, and thus $\phi(n+1|q+1) = 1$ for $\alpha < \beta$.

We can now rewrite (4) with only the nonzero terms in the summation as

$$\begin{aligned} &E[R[S+n+1]] - E[R[S+n]] \\ &= \sum_{q=q_0(n)}^{q_1(n)} \Pr(\varepsilon_q^{\mathcal{I}_{n-1}})[(1-\phi(n|q))\phi(n+1|q) \\ &\quad -\phi(n|q)(1-\phi(n+1|q+1))]. \end{aligned} \tag{5}$$

The probability $\Pr(\varepsilon_q^{\mathcal{I}_{n-1}})$ takes the form

$$\begin{aligned} \Pr(\varepsilon_q^{\mathcal{I}_{n-1}}) = &\sum_{i_1=0}^{n-q-1} \sum_{i_2=i_1+1}^{n-q} \cdots \sum_{i_q=i_{q-1}+1}^{n-1} \left[\prod_{a_1=0}^{i_1-1} (1-\phi(a_1|0))\right] \\ &\times \phi(i_1|0) \left[\prod_{a_2=i_1+1}^{i_2-1} (1-\phi(a_2|1))\right] \phi(i_2|1) \cdots \\ &\times \left[\prod_{a_q=i_{q-1}+1}^{i_q-1} (1-\phi(a_q|q-1))\right] \phi(i_q|q-1) \\ &\times \prod_{a_{q+1}=i_q+1}^{n-1} (1-\phi(a_{q+1}|q)). \end{aligned}$$

This expression comes from summing all combinations of choosing q of the n slots for there to be a request, and the term corresponding to slot a in which a request occurs takes the form $\phi(a|b)$. The other terms take the form $1-\phi(a|b)$.

We now define the inverse functions for $q_0(n)$ and $q_1(n)$ to determine for which values of a the terms of the summation above are nonzero. We define $q_0^{-1}(a) = \max\{n \in \mathbb{Z}_{\geq 0} : \tilde{p}[S] - \alpha(n-1) + \beta a > 1\}$. Then the terms in the above summation cannot be of the form $\phi(a|q)$ for $q < q_0^{-1}(a)$. Similarly, we define $q_1^{-1}(a) = \min\{n \in \mathbb{Z}_{\geq 0} : \tilde{p}[S] - \alpha(n+1) + \beta a < 0\}$. Then the terms in the above summation cannot be of the form $1-\phi(a|q)$ for $q > q_1^{-1}(a)$. Eliminating the terms that necessarily evaluate to zero, we now have

$$\begin{aligned} &\Pr(\varepsilon_q^{\mathcal{I}_{n-1}}) = \left[\prod_{a_0=0}^{q_0^{-1}(0)-1} (1-\phi(a_0|0))\right] \\ &\times \sum_{i_1=q_0^{-1}(0)}^{q_0^{-1}(1)-1} \sum_{i_2=q_0^{-1}(1)}^{q_0^{-1}(2)-1} \cdots \sum_{i_{q'}=i_{q'-1}+1}^{q_1(n-1)} \\ &\left[\prod_{a_1=q_0^{-1}(0)}^{i_1} (1-\phi(a_1|0))\right] \phi(i_1|0) \left[\prod_{b_1=i_1+1}^{q_0^{-1}(1)-1} (1-\phi(b_1|1))\right] \\ &\times \left[\prod_{a_2=q_0^{-1}(1)}^{i_2} (1-\phi(a_2|1))\right] \phi(i_2|1) \left[\prod_{b_2=i_2+1}^{q_0^{-1}(2)-1} (1-\phi(b_2|2))\right] \\ &\qquad \vdots \\ &\times \left[\prod_{a_{q'}=q_0^{-1}(q'-1)}^{i_{q'}} (1-\phi(a_{q'}|q'-1))\right] \phi(i_{q'}|q'-1) \\ &\times \left[\prod_{b_{q'}=i_{q'}+1}^{q_1(n-1)} (1-\phi(b_{q'}|q'))\right] \\ &\times \left[\prod_{a_{q'+1}=q_1(n-1)+1}^{n-1} \phi(a_{q'+1}|q-(n-a_{q'+1}))\right] \end{aligned} \tag{6}$$

where $q' = q - (n - q_1(n-1))$. In the last product term in this expression, the parameter $q-(n-a_{q'+1})$ in $\phi(\cdot|\cdot)$ ranges from $q-(n-q_1(n-1)-1)$ to $q-1$. If $q < n-q_1(n-1)-1$, the whole expression evaluates to zero.

Since in (5) the parameter q ranges from $q_0(n)$ to $q_1(n)$, we can declare the following sufficient condition for (5) to evaluate to zero:

$$\begin{aligned} n > &\left(\left\lfloor \frac{\alpha}{\beta}(n) - \frac{\tilde{p}[S]}{\beta} + \frac{1}{\beta} \right\rfloor\right)_0^n \\ &+ \left(\left\lfloor \frac{\alpha}{\beta}(n-1) - \frac{\tilde{p}[S]}{\beta} + \frac{1}{\beta} \right\rfloor\right)_0^{n-1} + 1. \end{aligned} \tag{7}$$

For sufficiently large n, the term $\left\lfloor \frac{\alpha}{\beta}(n) - \frac{\tilde{p}[S]}{\beta} + \frac{1}{\beta} \right\rfloor$ will be greater than zero, and since $\alpha < \beta$, it will be less than n. For large enough n, we then have the condition

$$\begin{aligned} n > &\left\lfloor \frac{\alpha}{\beta}(n) - \frac{\tilde{p}[S]}{\beta} + \frac{1}{\beta} \right\rfloor \\ &+ \left\lfloor \frac{\alpha}{\beta}(n-1) - \frac{\tilde{p}[S]}{\beta} + \frac{1}{\beta} \right\rfloor + 1. \end{aligned} \tag{8}$$

Finally, we remove the floor function for the two $\lfloor\cdot\rfloor$ terms, and we have another sufficient condition

$$n > \frac{\alpha}{\beta}(n) - \frac{\tilde{p}[S]}{\beta} + \frac{1}{\beta} + \frac{\alpha}{\beta}(n-1) - \frac{\tilde{p}[S]}{\beta} + \frac{1}{\beta} + 1 \tag{9}$$

since the right hand side (RHS) quantity is greater than or equal to that of (8). Solving for n and ensuring it is integer, we have

$$n > \left\lceil \frac{2(1-\tilde{p}[S]) - \alpha + \beta}{\beta - 2\alpha} \right\rceil. \tag{10}$$

for $1 - 2\alpha/\beta > 0$.

B. Numerical Examples

We provide some numerical examples of the request rate evolution in Figure 2. For $\alpha = 0.4$, $\beta = 0.9556$, $\tilde{p}[S] = 0.5$, the condition (10) evaluates to $n > 10$. For $\alpha = 0.1$, $\beta = 0.258$, $\tilde{p}[S] = 0.5$, the sufficient condition evaluates to $n > 20$. We observe in the plot that the request rate in the first case reaches its steady state sooner than in the second case, so the sufficient condition is useful for modeling the request profile of a system. We also plot the results for the case of $\alpha = 0.1$, $\beta = 0.245$, $\tilde{p}[S] = 0.5$, $1 - 2\alpha/\beta < 0$, so the sufficient condition does not apply (nor is it satisfied), but the plot appears to reach a steady state or at least approach an asymptote. So there are cases that do not satisfy the sufficient condition and which we are not yet able to analyze the behavior. We also note that the particular value at which the requests settle at in steady state varies depending on the parameters, and we can even achieve a value close to zero, as in the case of $\alpha = 0.99$, $\beta = 0.999$, $\tilde{p}[S] = 0.5$ (sufficient condition also does not apply here).

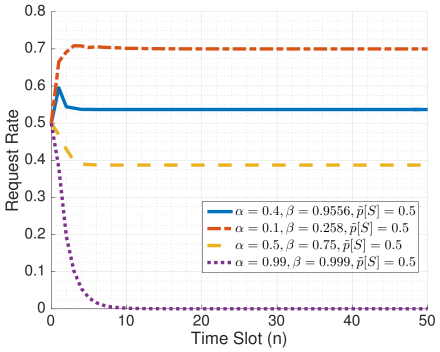

Fig. 2. Request rate vs. time slot.

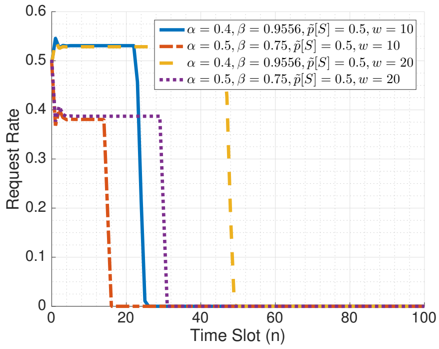

Fig. 3. Request rate vs. time slot.

V. Limiting the History of Requests

To compare the performance between caching policies over an infinite horizon, it would help if the request rate eventually decayed to zero. Otherwise, the missed request rate would continually escalate as more packets arrive to the system. In this section, we consider altering the proposed model to limit the history of requests by using a moving time window, so that only the recent requests in the window affect the request rate. The behavior of this model is that the request rate approaches zero for larger n, since the age grows without limit, but the requests are limited by the length of the window. To model this, we replace $r_m[S_m, S_m + n - 1]$ in the request rate expression with $r_m[S_m + n - w, S_m + n - 1]$, $w \in \mathbb{Z}_{\geq 0}$ being the length of the time window. We also note that $\tilde{p}_m[S]$ for $S > S_m$ is insufficient for tracking the evolution of the request rate, and the recent window of requests is needed.[2]

The request rate for a windowed request history is defined as

$$E[R_m[S+n]] = \begin{cases} \sum_{q=0}^{n} \phi_m(n|q) \Pr(\varepsilon_{m,q}^{\mathcal{I}_{n-1}}), & \text{if } 0 \leq n \leq w \\ \sum_{q=0}^{w} \phi_m(n|q) \Pr(\varepsilon_{m,q}^{[S+n-w, S+n-1]}), & \text{if } n > w \end{cases}$$

In this case of a windowed request history, a sufficient condition for when the request rate goes to zero is, for $n > w$,

$$q_0(n) > w$$

since $\phi_m(n|q)$ in the expression above will be zero for all $q \in [0, w]$ when n satisfies this condition. From this we can get another sufficient condition

$$n > \left\lceil \frac{\beta w + \tilde{p}[S]}{\alpha} \right\rceil$$

Due to the difficulty in tracking the moving window of requests, we simulate the request model and provide the results in Figure 3. We first consider two cases with a request history window of length $w = 10$. For $\alpha = 0.4$, $\beta = 0.9556$, $\tilde{p}[S] = 0.5$, the condition above yields $n > 26$, which is the actual slot that it reaches zero. For $\alpha = 0.5$, $\beta = 0.75$, $\tilde{p}[S] = 0.5$, the condition yields $n > 16$, and again is the actual slot that it reaches zero. Reducing the ratio α/β results in a faster decaying request model. We then increase the window $w = 20$ and simulate for the same parameters as the previous two. We see that using a longer request window results in a request model that reaches zero later.

VI. Conclusion

We studied a mobile content caching system and proposed a dynamic model of requests for content that incorporates the age and popularity of the information. Under this model, we have demonstrated that the optimal policy for minimizing the number of missed requests is to keep the packets that have the highest instantaneous request value in the cache. This policy does not depend on the exact age or history of requests, and we defined an effective age metric that is a sufficient statistic for optimal cache management. We also analyzed the request model and provided a sufficient condition for when the request rate reaches steady state. We also modified the request model in which the impact of the popularity is limited to a finite time window, and we provided a sufficient condition for when the request rate reaches zero. Future work includes further analysis of the modified model and optimal cache management, and studying a system in which arriving packets are updates to the cached content.

[2] The proof of optimality of policy $\tilde{\pi}$ (Theorem 1) may not apply exactly for this model, and we will consider the optimal policy in future work.

References

[1] Cisco Inc., "The Zettabyte era: Trends and Analysis," White Paper, June 2016.
[2] M. Dehghan, L. Massouli, D.Towsley, D.S. Menasch, and Y. C. Tay, "A utility optimization approach to network cache design," In *Proc. IEEE INFOCOM*, 2016.
[3] P. Rodriguez, C. Spanner, and E.W. Biersack, "Analysis of Web Caching Architectures: Hierarchical and Distributed Caching," IEEE Trans. Networking, vol. 9, no. 4, pp. 404–418, Aug. 2001.
[4] J. Zhang, "A Literature Survey of Cooperative Caching in Content Distribution Networks," arXiv:1210.0071v1 Sep. 2012.
[5] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," In *Proc. IEEE INFOCOM*, Mar. 2012.
[6] M. Dong, H. Li, K. Ota, and J. Xiao, "Rule caching in SDN-enabled mobile access networks," IEEE Network, vol. 29, no. 4, pp. 40–45, Jul. 2015.
[7] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," In *Proc. IEEE INFOCOM*, Mar. 2010.
[8] V. S. Mookerjee and Y. Tan, "Analysis of a Least Recently Used Cache Management Policy for Web Browsers," Operations Research, vol. 50, no. 2, pp. 345–357, Mar. 2002.